Local features are widely utilized in a large number of applications, e.g., object categorization, image retrieval, robust matching, and robot localization. In this review, we focus on detectors and local descriptors. Both earlier corner detectors, e.g., Harris corner detector, and later region detectors, e.g., Harris affine region detector, are described in brief. Most kinds of descriptors are described and summarized in a comprehensive way. Five types of descriptors are included, which are filter-based descriptors, distribution-based descriptors, textons, derivative-based descriptors and others. Finally, the matching methods and different applications with respect to the local features are also mentioned. The objective of this review is to provide a brief introduction for new researchers to the local feature research field, so that they can follow an appropriate methodology according to their specific requirements.

© 2008 Elsevier B.V. All rights reserved.

1. Introduction

Nowadays, local features (local descriptors), which are distinctive and yet invariant to many kinds of geometric and photometric transformations, have been gaining more and more attention because of their promising performance. They are being applied widely in computer vision research, for such tasks as image retrieval [41,63,73,74,78], image registration [5], object recognition [9,10,30–32,42,50,72], texture classification [29,37,40,81,82], robot localization [65], wide baseline matching [22,45,56,61,75,78], and video shot retrieval [67,68].

There are two different ways of utilizing local features in applications: (i) traditional utilization, which involves the following three steps: feature detection, feature description, and feature matching; (ii) bag-of-features [55] and hyperfeatures [1], which include the following four steps: feature detection, feature description, feature clustering, and frequency histogram construction for image representation. The focus of this review is on local features. A local feature consists of a feature detector and a feature descriptor.

Feature detectors can be traced back to the Moravec’s corner detector [52], which looks for the local maximum of minimum intensity changes. As pointed by Harris and Stephens [31], the response of this detector is anisotropic, noisy, and sensitive to edges. To reduce these shortcomings, the Harris corner detector [28] was developed. However, it fails to deal with scale changes, which always occur in images. Therefore, the construction of detectors that can cope with this scaling problem is important. Lowe [42] pioneered a scale invariant local feature, namely the scale invariant feature transform (SIFT). It consists of a detector and a descriptor. The SIFT’s detector finds the local maximums of a series of difference of Gaussian (DoG) images. Mikolajczyk and Schmid [47] developed the Harris–Laplace detector by combining: (i) the Harris corner detector and (ii) the Laplace function for characteristic scale selection, for scale invariant feature detection. To deal with the viewpoint changes, Mikolajczyk and Schmid [47] put forward the Harris–Laplace detector by combining: (i) the Harris corner detector and (ii) the Laplace function for characteristic scale selection, for scale invariant feature detection. To deal with the viewpoint changes, Mikolajczyk and Schmid [47] put forward the Harris–Laplace detector and (ii) the Laplace function for characteristic scale selection, for scale invariant feature detection. To deal with the viewpoint changes, Mikolajczyk and Schmid [47] put forward the Harris–Laplace detector by combining: (i) the Harris corner detector and (ii) the Laplace function for characteristic scale selection, for scale invariant feature detection.
To represent points and regions, which are detected by the above methods, a large number of different local descriptors have been developed. The earliest local descriptor could be the local derivatives [36]. Florack et al. [23] incorporated a number of local derivatives and constructed the differential invariants, which are rotational invariant, for local feature representation. Schmid and Mohr [63] extended local derivatives as the local grayscale invariants for image retrieval, Freeman and Adelson [25] proposed steerable filters, which are linear combinations of a number of basis filters, for orientation and scale selection to handle tasks in image processing and computer vision research. Marcelja [44] and Daugman [17,18] modeled the responses of the mammalian visual cortex through a series of Gabor functions [17,18], because these functions can suitably represent the receptive field profiles in cortical simple cells. Therefore, Gabor filters can be applied for local feature description. Wavelets, which are effective and efficient for multiresolution analysis, can also represent local features. Textons, e.g., 2D textons, 3D textons [40], and the Varma–Zisserman model [81], have been demonstrated to have good performance for texture classification. A texton dictionary is constructed from a number of textures and a clustering algorithm is applied to select a small number of models to represent each texture. Texton representation is also a good choice for local feature modeling. Van Gool et al. [80] computed them for moments up to second order and second basis degree based on the derivatives of x and y directions of an image patch. Past research has shown the effectiveness of the SIFT descriptor, which is a 3D histogram for gradient magnitudes and orientations representation. This feature is invariant to changes in partial illumination, background clutter, occlusion, and transformations in terms of rotation and scaling. Schaffalitzky and Zisserman [61] introduced complex filters for generating kernel functions for efficient multi-view matching. Shape context, developed by Belongie et al. [9], describes the distribution of the rest points of a shape with respect to a reference point on the shape for matching. Based on the phase and amplitude of steerable bandpass filters, Carneiro and Jepson [12] proposed phase-based local features, which improve invariance to illumination changes. The spin image, originally developed by Johnson and Hebert [32] for 3D object recognition, has been applied to texture modeling by Lazebnik et al. [37], Ke and Sukthankar [34] simplified the SIFT descriptor by utilizing principal component analysis (PCA) to normalized gradient patches to achieve fast matching and invariance to image deformations. This method is named as PCA-SIFT. Lazebnik et al. [37] put forward the rotation invariant feature transform (RIFT), which divides each circular normalized patch into concentric rings, each of which is associated with a gradient orientation histogram. A recent study reports the significance of the gradient location and orientation histogram (GLOH), proposed by Mikolajczyk and Schmid [48], which is an extension of the SIFT descriptor. Similar to the PCA-SIFT, GLOH also applies PCA to reduce the dimension of the descriptor. All the preceding descriptors do not consider color information, which can be important in visual representations. To this end, Van De Weijer and Schmid [79] developed four color descriptors, which are histograms of RGB, hue, opponent angle, and spherical angle. Preliminary experiments have demonstrated the effectiveness of these descriptors.

Feature matching is an important step to measure the similarity or dissimilarity between two images, which are represented by two sets of local features, where a similarity metric is constructed based on the correspondences of the local features. In most applications, the following three matching methods are applied: (i) threshold-based matching, (ii) nearest neighbor matching, and (iii) nearest neighbor distance ratio matching. Threshold-based matching finds all possible candidate points in other image for each point in the reference image, in case that the distance between the descriptors of the candidate point and the reference point is below a specified threshold. Nearest neighbor matching algorithms find the point with the closest descriptor to a reference point. Nearest neighbor distance ratio matching utilizes the ratio between the distance of the nearest and the second-nearest neighbors for a reference point. Using which form of matching method depends on a specific application. If a simple and fast strategy is required, the threshold-based matching is often the best choice; if an accurate and effective algorithm is a prerequisite, the nearest neighbor distance ratio matching has distinct advantages. For detector [49,53,64] and descriptor [48,53] performance evaluations, almost all existing works are based on one of these matching methods.

Existing detector performance evaluations [49,53] have demonstrated that: (i) under viewpoint changes, MSER outperform others for both structured and textured images; (ii) under scale changes, Hessian affine detector achieves the best results for both structured and textured images; (iii) under image blur, MSER performs poorly for structured flat images but generally reliable for textured images; others work similarly for both; (iv) under JPEG compression for structured images, Hessian and Harris affine perform best; (v) under illumination changes, all detectors perform well and MSER works best; and (vi) Hessian affine and DoG consistently outperform others for 3D objects.

Previous descriptor evaluations [48,53] revealed that: (i) under rotation changes, all descriptors have a similar performance for structured images. GLOH, SIFT and shape context obtain the best results for textured images although all descriptors do not perform well for this kind of images; (ii) under viewpoint changes, the results are better for textured images than for structured image: the GLOH descriptor perform best for structured images and SIFT obtains the best results for textured images; (iii) under scale changes, SIFT and GLOH obtain the best results for both textured and structured images; (iv) under image blur, the performances of all descriptors are degraded significantly; GLOH and PCA-SIFT obtain the best results; (v) for JPEG compression of structured images, when a high false positive rate is allowable, SIFT works best. Otherwise, PCA-SIFT is the best choice; and (vi) under illumination changes, GLOH performs best for illumination normalized regions. In summary, GLOH obtains the best results, closely followed by SIFT, which is always regarded as state of the art and has been demonstrated to achieve the most stable and efficient results.

The organization of this paper is as following. Detectors are reviewed in Section 2; local descriptors are described in Section 3; and Section 4 concludes.

2. Feature detection

Feature detection is the requisite step in obtaining local feature descriptions. It locates points and regions, and is generally capable of reproducing similar levels of performances to human observers in locating elemental features in a wide range of image types. Most of the existing detectors can be categorized into two types: (i) corner detectors and (ii) region detectors.

2.1. Definitions

If an image is defined as a function \( I(\mathbf{p}) \), where the domain of \( I \) is the set of locations \( \mathbf{p} = [x,y] \) for pixels. The derivative of \( I(\mathbf{p}) \) with respect to \( x \) is given by \( I_x(\mathbf{p}) \) or \( I_x \). The derivative of \( I(\mathbf{p}) \) with respect to \( y \) is given by \( I_y(\mathbf{p}) \) or \( I_y \). A Gaussian kernel with a local
scale parameter \( \sigma \) is defined as \( g(\vec{p}; \sigma) = (1/2\pi\sigma^2) \exp(-|\vec{p}|^2/(2\sigma^2)) \).

We introduce the following definitions relevant to the detectors in this paper.

**Definition 1 (Scale Space).** Scale space representation is to characterize an image at different scales. It is universally applied for feature detection and image matching. For a given image \( I(\vec{p}) \), the corresponding linear scale space representation is a series of responses \( L(\vec{p}; \sigma) \), which is obtained by convoluting \( I(\vec{p}) \) with \( g(\vec{p}; \sigma) \), i.e.,

\[
L(\vec{p}; \sigma) = \int I(\vec{p} - \vec{q})g(\vec{q}; \sigma) \, d\vec{q} = I(\vec{p})g(\vec{p}; \sigma).
\]

**Definition 2 (Harris Matrix, Second Moment Matrix, or Structure Tensor).** The Harris matrix, a matrix of partial derivatives, is typically utilized to represent the gradient information in the area of image processing and computer vision research. It was defined in Harris corner detector and the eigenvalues of this matrix determine whether or not a point is a corner. For a given image \( I(\vec{p}) \), the Harris matrix is defined by

\[
A = \nabla I \otimes \nabla I = \begin{bmatrix} I_x & I_y \\ I_y & -I_x \end{bmatrix}.
\]

**Definition 3 (Hessian Matrix).** The Hessian matrix of a given image \( I(\vec{p}) \) is the matrix of second partial derivatives with respect to \( \vec{p} \), i.e.,

\[
H = \nabla^2 I = \begin{bmatrix} I_{xx} & I_{xy} \\ I_{yx} & I_{yy} \end{bmatrix}.
\]

**Definition 4 (Laplacian Function).** The Laplacian operator is invariant to rotation in a given image \( I(\vec{p}) \). It is related to scale space representation and can be defined with respect to \( \vec{p} \) by

\[
\Delta I = \nabla^2 I = (\nabla \cdot \nabla)I = I_{xx} + I_{yy}.
\]

It is commonly used for blob detection and usually results in strong positive responses for dark blobs and strong negative responses for bright blobs with similar sizes.

**Definition 5 (Multi-scale Second Moment Matrix).** The multi-scale second moment matrix of a given image \( I(\vec{p}) \) can be defined as

\[
\Gamma(\vec{p}; \sigma_0, \sigma_1) = \int L_2(\vec{p}; \sigma_0) \, \nabla L_1(\vec{p}; \sigma_0) \, \nabla L_1(\vec{p}; \sigma_0) \, g(\vec{q}; \sigma_1) \, d\vec{q} = \begin{bmatrix} L_1^2(\vec{p}; \sigma_0) & L_1(\vec{p}; \sigma_0) L_2(\vec{p}; \sigma_0) \\ L_1(\vec{p}; \sigma_0) L_2^T(\vec{p}; \sigma_0) & L_2(\vec{p}; \sigma_0) L_2(\vec{p}; \sigma_0) \end{bmatrix} g(\vec{q}; \sigma_1),
\]

where \( L_x \) and \( L_y \) denote the partial derivatives of \( L(\vec{p}; \sigma_0) \) with respect to \( x \) and \( y \) in scale \( \sigma_0 \) (differentiation scale), respectively; and the function \( g(\vec{q}; \sigma_1) \) is a Gaussian kernel with scale \( \sigma_1 \) (integration scale). Obviously, \( \sigma_0 \) is for scale space smoothing before computing the image derivatives and \( \sigma_1 \) is for accumulating non-linear operations on derivative operators onto an integrated image descriptor.

**Definition 6 (Geometric Transformations).** Geometric transformations consist of translation, reflection, rotation, skew, scale, glide—all of which maintain the previous shape; while stretch, shear, enlargement and other topological transformations change the original shape. In a geometric transformation, each pixel at location \( \vec{p}_1 \) in a reference image is mapped to its corresponding location \( \vec{p}_2 \) in the target image. Different geometric transformations can be expressed by the same function with different transformation matrix \( A \) and translation vector \( \vec{b} \), i.e.,

\[
\vec{p}_2 = A \vec{p}_1 + \vec{b}.
\]
According to Refs. [28,52], the shortcomings of this detector are: (i) the response is anisotropic due to the consideration of shifts only at every 45°; (ii) the response is noisy due to the use of a binary rectangular window; and (iii) the detector responds to edges easily due to the consideration of only the minimum intensity changes.

2.2.2. Harris corner detector

Harris corner detector, also called the Plessey corner detector, was proposed by Harris and Stephens [28] to reduce the weaknesses of the Moravec’s corner detector, by: (i) perform an analytic expansion about the shift origin to involve all possible small shifts; (ii) substitute a smooth circular window for the binary rectangular one; and (iii) redevelop the corner response metric to take into account the intensity changes with the direction of shifts, respectively. It can determine whether a pixel is a corner, a point is on an edge, or a point is in a uniform intensity region; (ii) if \( \lambda_1 \approx 0 \) and \( \lambda_2 \approx 0 \), the pixel is on an edge; and iii) if \( \lambda_1 \) and \( \lambda_2 \) are both large, a corner is indicated. To reduce the computational complexity, the Harris corner metric [28] \( m_b = \det(A) - \kappa \text{tr}^2(A) \) replaces the eigenvalue computation for justification: (i) if \( m_b \) is small, the pixel is in a uniform intensity region; (ii) if \( m_b < 0 \), the pixel is on an edge; and (iii) if \( m_b > 0 \), the pixel is a corner. The detected Harris points are invariant to rotation. This detector finds locations with large gradient in all directions at a predefined scale, including corners. This detector fails to satisfactorily deal with scaling.

2.2.3. SUSAN

Smith and Brady [69] considered that pixels in a relatively small region are uniform in terms of brightness if these pixels belong to the same object. Based on this point of view, the SUSAN is implemented by comparing brightness within a circular mask \( C \). In detail, the brightness of each pixel within the mask, \( B \in C \), is compared with that of the nucleus \( B_0 \) (the center of \( C \)) by a comparison function:

\[
m_B = \exp(-t^{-1}(l(B) - l(B_0))^6),
\]

where \( t \) is a threshold. A nucleus \( B_0 \) is a corner, if the number of pixels, which are similar to \( B_0 \) in terms of brightness, is less than a given threshold. The SUSAN detector has good reliability. It locates corners accurately and is excellently insensitive to noise. It is relatively fast. However, it performs poorly for blurred images.

2.2.4. Trajkovic operator

To reduce computational complexity, a fast corner detector is developed by Trajkovic and Hedley [77]. The chief concept is to investigate the variations of intensity along arbitrary lines passing through a point within a neighborhood of the point, as shown in Fig. 3. Horizontal and vertical intensity variations are computed as

\[
\begin{align*}
r_A &= (f_A - f_C)^2 + (f_K - f_C)^2, \\
r_B &= (f_B - f_C)^2 + (f_K - f_C)^2, \\
r_C &= (f_K - f_C)^2 - (f_A - f_C)^2 - (f_B - f_C)^2 - (f_y - f_C)^2, \\
r_D &= (f_y - f_C)^2 - (f_A - f_C)^2 - (f_B - f_C)^2 - (f_y - f_C)^2.
\end{align*}
\]

where \( f_A, f_B, f_K, f_C \) and \( f_y \) are pixel values at the locations \( A, B, K, C \) and \( Y \), respectively. Then, the corner metric is computed as

\[
R = \min(R_A, R_B, R_C, R_D).
\]

For linear interpixel approximation, define \( Y = \min(Y_1, Y_2) \) and \( X = r_B - r_A - 2Y \), where

\[
\begin{align*}
Y_1 &= (f_B - f_A)(f_K - f_C) + (f_B - f_K)(f_B - f_C), \\
Y_2 &= (f_B - f_K)(f_A - f_C) + (f_B - f_A)(f_B - f_C), \\
Z &= r_A,
\end{align*}
\]

the minimum corner response is \( R_Z = Z - Y^2 / X \) with the constraints \( Y < 0 \) and \( X + Y > 0 \); for circular interpixel approximation, define \( X = (r_A - r_B) / 2 \) and \( Y = \min(Y_1, Y_2) \), where

\[
\begin{align*}
Y_1 &= (f_A - f_C) \cdot (f_B - f_C) + (f_B - f_C) \cdot (f_B - f_C) + (f_B - f_C) \cdot (f_B - f_C), \\
Y_2 &= (f_A - f_C) \cdot (f_B - f_C) + (f_A - f_C) \cdot (f_B - f_C) + (f_B - f_C) \cdot (f_B - f_C),
\end{align*}
\]

the maximum corner response is \( R_Z = Z - \sqrt{X^2 + Y^2} \) with the constraint \( B < 0 \).

The fast corner detection is a three-step algorithm:

1. the corner measure \( R_0 \) is computed at every pixel in a low resolution version of the image. Each pixel with its response higher than a predefined threshold \( T_1 \) is considered as a “potential corner”;

2. the corner responses in the “potential corner” region are averaged to produce the final corner measure \( R_1 \);

3. corner points are found by thresholding the final corner responses \( R_1 \).
(2) Using the full-resolution image, for each potential corner:
   (i) Compute $R_0$. If the response is higher than a second
       threshold $T_2$, perform (ii); otherwise, the pixel is not a
       corner point and no further computation is needed;
   (ii) Use $R_l(R_c)$ to compute a new response. If the response
        is higher than $T_2$, then conduct (3); otherwise, the pixel is
        not a corner point.
(3) Pixels with a locally maximal $R_l(R_c)$ are selected as corners,
    which is called non-maximum suppression.

Because the number of corners is usually a small fraction of the
image pixels, it is not meaningful to apply $R_l(R_c)$ to each pixel in
the image due to most low responses. Therefore, by firstly using $R_0$
for each pixel, the fast corner detector largely reduces the
computational complexity and thus is very fast. A high response,
$R_l(R_c)$ verifies the existence of a corner.

2.2.5. High-speed corner detector

Rosten and Drummond [59] utilized machine learning to speed
up the corner detection. The process includes the following three
stages:

1. Segment test on a Bresenham circle of a center pixel $p$ with
   radius 3: this step is computationally efficient to remove most
   of non-corner candidates. It is based on a logistical test, i.e., $p$
   is not a corner if a pixel with position $x$ and another pixel with
   position $x+8$ are similar to $p$ in terms of intensity. The test will
   be conducted on 12 consecutive positions;
2. Classification-based corner detection: apply ID3 tree classifier
   [58] to determine whether $p$ is a corner based on 16 features.
   Each feature is 0, 1, or −1. If a pixel with position $x$ on the
   Bresenham circle of $p$ is larger (smaller) than $p$, the
corresponding feature is 1 (−1). Otherwise, the feature is 0; and
3. Corner verification: the non-maximum suppression is utilized
   for verification.

2.2.6. Others detectors in the timeline

The Beaudet corner detector [7] is a rotational-invariant
operator based on a corner measure, i.e., the determinant of the
Hessian matrix. This operator is sensitive to noise because of the
computation of second derivatives. Kitchen and Rosenfeld [35]
detect corners at the local maximum of a corner measure based
on the change of gradient direction along an edge weighted by the
gradient magnitude. Similar to the Harris corner detector, the
Förstner operator [24] also utilizes a corner measure defined by
the second moment matrix. The threshold is determined by local
statistics. Nobel [54] modified the Harris corner measure as

$$m_n = \text{tr}(A) + \epsilon^{-1} \text{det}(A).$$  

Moreover, the differential geometry of a facet model was
applied to represent 2D surface features, including corners.
Deriche and Giraudon [19] generalized the Beaudet operator for
two scales with the zero crossing of the Laplacian image used to
obtain an accurate position of a corner. By analyzing the optical
flow equation proposed by Lucas and Kanade [43], Tomasi and
Kanade [76] obtained the relationship:

$$\begin{bmatrix} \sum w_{x} & \sum w_{x} l_{x} \\ \sum w_{x} l_{x} & \sum w_{x} l_{x} l_{x} \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = - \begin{bmatrix} \sum w_{x} l_{y} \\ \sum w_{x} l_{x} l_{y} \end{bmatrix}. \tag{11}$$

Here, an image is described by $(x, y, t)$, where $x$ and $y$ are the
space variables, $t$ is the time variable and $w$ is a weighting
function. Corners are chosen as image locations where the
condition is satisfied. Shi and Tomasi [66] proposed a new corner
detection algorithm that is strongly based on the Harris corner
detector, arguing that $\min(x, y)$ is a better choice than
$m_n = \text{det}(A) - \kappa \text{tr}^2(A)$. Chan et al. [13] introduced wavelets for
corner detection because wavelets provide multi-resolution
(multi-scale) analysis. Corners are detected by a given threshold
on a high–high component of an input image, decomposed by a
B-spline wavelet in several scales. Chen et al. [14] replaced the
B-spline wavelet with the Gaussian function based wavelet for
corner detection. Quddus and Fabiny [57] modified Gabor wavelet
for corner detection. Recently, Gao et al. [26] applied a different
corner measure for Gabor wavelet based corner detection. Wang
and Brady [83] designed a corner detector for motion estimation,
with an image regarded as a surface and the corner metric $m_n = 
\sqrt{\lambda_1 - \lambda_2} \lambda_2$ applied to detect corners. To reduce the computa-
tional complexity of the Harris corner detector, Zheng et al. [84]
proposed a gradient-direction corner detector based on a
simplified corner measure constructed from the Harris corner
corners. Ando [2] detected corners and edges based on gradient
covariance matrix and gradient projection. Sojka [70] applied a
measurement function, where the weighting coefficients are
computed based on the Bayes’ theorem, to justify the variance of
the gradient directions.

Both the Harris corner detector and the SUSAN algorithm are
reliable, but the latter greatly outperforms the former on
localization accuracy and speed. According to Ref. [51], the
Kitchen and Rosenfeld detector is the fastest compared with the
Harris corner detector and the SUSAN. However, it is worth
emphasizing that almost all the corner detectors discussed above
can deal with affine transformations.

2.3. Region detectors

In this subsection, we review the following region detectors:
(i) the Harris–Laplace region detector [47], (ii) the Harris affine
and Hessian affine corner detector [47], (iii) the edge-based region
detector [78], (iv) the intensity-based region detector [78], (v) the
MSER detector [45] and (vi) the salient region detector [33] as
important examples of recently developed region detectors.

2.3.1. Harris–Laplace

Harris–Laplace [47] region detector locates potentially relevant
points, interest points, with the Harris corner detector and then
selects the point with a characteristic scale, which is the
extremum of the Laplacian over different scales. To describe the
Harris–Laplace detector, we need to define the multi-scale Harris
corner measure and the scale-normalized Laplacian operator.

Multi-scale Harris corner measure is given by

$$m_n(p; \sigma_d, \sigma_l) = \text{det}(\Gamma(p; \sigma_d, \sigma_l)) - \kappa \text{tr}^2(\Gamma(p; \sigma_d, \sigma_l)),$$  

where $\Gamma(p; \sigma_d, \sigma_l)$ is the multi-scale second moment matrix; $\kappa$ is a
predefined scalar; and $\sigma_d$ and $\sigma_l$ are local scale parameters.

With a given scale parameter $\sigma$, the scale-normalized Laplacian
operator is

$$\Delta_n \Lambda(p; \sigma) = \sigma \Delta \Lambda(p; \sigma) + (\Lambda_{xx}(p; \sigma) + \Lambda_{yy}(p; \sigma)),$$  

where $\Lambda_{xx}$ and $\Lambda_{yy}$ are second partial derivatives with respect to $x$
and $y$, respectively.

Based on the two definitions above, the Harris–Laplace
detector contains two steps:

1. Pixels located at the spatial maximum of $m_n(p; \sigma_d, \sigma_l)$ are
   selected as candidate interest points, i.e.,
   $$\hat{p} = \text{arg max}_p m_n(p; \sigma_d, \sigma_l), \quad \sigma_l = \gamma^2 \sigma_d;$$  

2. a characteristic scale is selected for each candidate interest point at the local extremum over scales of $A_N \ell(p; \sigma_D)$, i.e.,

$$\sigma_D = \arg \min_{\sigma_D} A_N \ell(p; \sigma_D).$$  \hspace{1cm} (15)

2.3.2. Harris affine and Hessian affine

Mikolajczyk and Schmid [47] proposed the Harris affine and Hessian affine detectors. The process is as follows:

1. detect interest points using the multi-scale Harris corner measure, defined in Eq. (12), or the Hessian matrix based detector;
2. select the characteristic scale for each interest point. This step is the same as the second step for scale selection in the Harris–Laplace detector;
3. determine the shape associated with an interest point by the eigenvalues and eigenvectors of the second moment matrix $A$;
4. normalize the shape into a circle according to $p = A^{-1/2} \tilde{p}$; and
5. conduct steps 2–4 for interest points.

2.3.3. Edge-based region detector

Edge-based region detector [78] is chiefly based chiefly on the following two motivations: (i) edges are stable under affine transformations; (ii) edge-based region detection is more effective than corner-based region detection. Edges are further classified as curved edges or straight edges.

For curved edges, the following steps are applied for region detection:

1. detect interest points $\tilde{p}$ by the Harris corner detector;
2. extract two nearby edges from $\tilde{p}$ to $\tilde{p}_1$ and $\tilde{p}_2$, respectively, through the Canny edge detector [11], as shown in Fig. 4;
3. the two points $\tilde{p}_1$ and $\tilde{p}_2$ move away from $\tilde{p}$ with the relative speeds, which are related to the equality of relative affine invariant parameters $l_1$ and $l_2$:

$$l_i = \int \left( \frac{\partial \ell_i(s_i)}{\partial s_i} \right) (p - \tilde{p}(s_i)) \, ds_i, \quad i = 1, 2,$$  \hspace{1cm} (16)

where $s_i$ is an arbitrary curve parameter. For each value $l$ ($l_1 = l_2$), the two points $\tilde{p}_1$ and $\tilde{p}_2$, associated with the corner $\tilde{p}$, define a region $\Omega(l)$, a parallelogram, which is spanned by the vectors $\tilde{p}_1 - \tilde{p}$ and $\tilde{p}_2 - \tilde{p}$; and
4. the points stop when a specific photometric quantity of the texture covered by $\Omega(l)$ achieves an extremum. The following measures are suggested:

$$f_1(\Omega) = \frac{m_{10}^2}{m_{00}^3},$$  \hspace{1cm} (17)

$$f_2(\Omega) = \frac{\det([\tilde{p}_1 - \tilde{p}_2, \tilde{p}_2 - \tilde{p}_1])}{\det([\tilde{p} - \tilde{p}_1, \tilde{p} - \tilde{p}_2])} \frac{m_{10}^1}{\sqrt{m_{00}^2 m_{00}^6 - (m_{10}^1)^2}},$$  \hspace{1cm} (18)

$$f_3(\Omega) = \frac{\det([\tilde{p} - \tilde{p}_1, \tilde{q} - \tilde{p}_2])}{\det([\tilde{p} - \tilde{p}_1, \tilde{p} - \tilde{p}_2])} \frac{m_{10}^1}{\sqrt{m_{00}^2 m_{00}^6 - (m_{10}^1)^2}},$$  \hspace{1cm} (19)

where $m_{00}^6 = \int_0^1 F(x,y)^2 \, dx \, dy$, $\tilde{p}_2 = [m_{10}^1/m_{00}^6, m_{01}^1/m_{00}^6]^T$, and $\tilde{q}$ is the corner of the parallelogram opposite to $\tilde{p}$, as shown in Fig. 4.

In the case of straight edges, the following steps are applied for region detection:

1. combine $f_2(\Omega)$ and $f_3(\Omega)$ together; and
2. take the intersection of the valleys, which are minima of the two functions above, to estimate the parameters $s_1$ and $s_2$ along the straight edges.

2.3.4. Intensity-based region detector

The intensity-based region detector [78] is based on intensity extrema detected at multi-scales. The procedure is as follows:

1. the local intensity extrema are detected using non-maximum suppression, as shown in the part (a) of Fig. 5;
2. given such an extremum, the intensity function along rays emanating from the extremum is

$$f_1(t) = ||t(t) - I_0|| \left( \max \left( t^{-1} \int_0^t ||t(t) - I_0|| \, dt \right) \right)^{-1},$$  \hspace{1cm} (20)

where $t$ is the Euclidean arclength along a ray; $I(t)$ is the intensity at position $t$; $I_0$ is the intensity extremum; and $d$ is a small number to avoid to be divided by zero. This step is described in the part (b) of Fig. 5;
3. all points corresponding to the maximum of the intensity function along rays emanating from the same local extremum are linked together to enclose an affine region. Usually, the shape is irregular, as shown in the part (c) of Fig. 5;
4. an ellipse, which has the same shape moments as the irregular region, is used to describe (replace) it to ensure affine invariance, as shown in the part (d) of Fig. 5; and
5. the ellipse is enlarged with a ratio $2$, as shown in the part (e) of Fig. 5. It is to make the detected region more distinctive and therefore speed up the matching procedure.

2.3.5. Maximally stable extremal region detector

The MSER detector was proposed by Matas et al. [45]. The detection procedure, which is similar to the watershed-based image segmentation, is described by the following steps:

1. an image is thresholded into a series of binary images, from white to black. The thresholds are in an ascending order;
2. blobs corresponding to local intensity minimum will appear and merge at some point. All closed components in the image are extremal regions; and
3. only the extremal regions that remain unchanged over a range of thresholded images are selected as MSERs.

MSERs have the following properties: (i) all intensities in each MSER are either lower (dark extremal region) or higher (bright extremal region) than intensities outside its boundary; and (ii) each MSER is affine invariant for both geometrical and photometrical transformations. It should be borne in mind that the output of the detector is not a binary image.
2.3.6. Salient region detector

Kadir and Brady [33] proposed the salient region detector, which is based on the probability density function (pdf) of intensity values \( p(l) \) computed over an elliptical region \( \Omega \). The detailed procedure is as follows:

1. At each pixel \( \beta \), calculate the entropy of the pdf \( p(l) \) over three parameters \((s, \theta, \lambda)\) of an ellipse centered at \( \beta \), where \( s \) is the scale; \( \theta \) is the orientation; and \( \lambda \) is the ratio of major axes.
2. The set of entropy extrema over scales and \((s, \theta, \lambda)\) are recorded as candidate salient regions. The entropy \( H \) is
   \[
   H = - \sum_l p(l) \log p(l),
   \]
   (21)
3. The candidate salient regions are ranked according to the magnitude of the derivative of the pdf \( p(l; s, \theta, \lambda) \) for each extremum with respect to \( s \), which is described by
   \[
   w = \frac{s^2}{2s - 1} \sum_l \left| \frac{\partial p(l; s, \theta, \lambda)}{\partial s} \right|. 
   \]
   (22)
4. Top \( P \) ranked regions are retained by ranking their saliency \( y = \text{Hw} \) in \( \Omega \).

However, in this detector, images have to be manually marked to select the correct correspondences. This step is not practical in most real-world applications.

2.3.7. Difference of Gaussian operator and sift detector

DoG method was proposed by Crowley and Parker [15]. It selects the scale-space extrema in a series of DoG images by convoluting an image with DoG functions (with different local scales). The detected points are candidate keypoints.

Lowe [42] extended this operator to deal with scale changes for the SIFT, which is specifically designed for scaling. The extension includes the following three steps: (i) the scale-space extrema detection, (ii) the keypoint localization, and (iii) the orientation assignment. The first step is the same as the DoG operator. In the second step, low contrast candidate points and edge response points along an edge are discarded. This step ensures that the keypoints are more stable for matching and recognition. The final step assigns the dominant orientation to a keypoint. In detail, the following sub-steps are applied to achieve the above objectives.

For a given image \( I_0(\beta) \), the scale-space extrema detection has the following sub-steps:
1. To reserve the highest spatial frequencies, i.e., to increase the number of stable keypoints, the linear interpolation is utilized to double the size of a given image: \( I_0(\beta) \rightarrow I_1(\beta) \);
2. Construct the scale space \( L(\beta; \sigma) \) with a series of \( g(\beta; \sigma) \), where \( 1 \leq i \leq N \) and \( N \gg 4 \). Moreover, \( \sigma_i = 2^{i-1}(N-1)\sigma_1 \), and \( \sigma_1 \) is a predefined value, e.g., \( \sigma_1 = 1.6 \);
3. Obtain DoG images \( D_j(\beta) = L(\beta; \sigma_{j+1}) - L(\beta; \sigma_j) \), for \( 1 \leq j \leq N-1 \);
4. Find the local extrema of \( D_j(\beta) \) for \( 2 \leq k \leq N-2 \). A point \( \tilde{\beta} = [x, y]^T \) in \( D_j(\beta) \) is a local extremum if and only if \( D_k(\tilde{\beta}) \) is the largest or the smallest among its 26 neighbors, i.e., 8 neighbors in \( D_0 \) and 9 neighbors in adjacent scales below \( D_{k-1} \) and above \( D_{k+1} \);
5. Resample \( L(\tilde{\beta}; \sigma_1) \), where \( \sigma_1 = 2\sigma_1 \), by taking every second pixel in each row and column. Conduct steps 2–4; and
6. Conduct step 5 many times.

For the keypoint localization, we have the following sub-steps:
1. Reject extrema which have low contrast, thus are sensitive to noise, by calculating
   \[
   \tilde{z}_e = \left[ \begin{array}{c} x_e \\ y_e \\ \sigma_e \\ \frac{\partial^2 L}{\partial x^2} \\ \frac{\partial^2 L}{\partial y^2} \\ \frac{\partial^2 L}{\partial x \partial y} \\ \frac{\partial^2 L}{\partial x \sigma} \\ \frac{\partial^2 L}{\partial y \sigma} \\ \frac{\partial^2 L}{\partial \sigma^2} \end{array} \right]. 
   \]
   (23)
2. Conduct the theoretical estimation \( \tilde{z}_e \) with the following sub-steps:
   \[
   L(\tilde{z}_e) = L(\tilde{z}) + \frac{1}{2} \tilde{z}^T \partial^2 L \tilde{z}_e. 
   \]
   (24)

The derivatives are approximated by simple differences, e.g.,
\[
\partial L = \frac{L(\tilde{\beta} + \sigma_1) - L(\tilde{\beta} - \sigma_1)}{2}. 
\]
3. The orientation assignment is applied to create keypoints based on each extreme \( \tilde{z}_m = [x_m, y_m, \sigma_m]^T \) with the following sub-steps:
1. Calculate the magnitude and the orientation of all points in the circular region of an extreme:
   \[
   m(x, y, \sigma) = \sqrt{(L(x + 1, y; \sigma) - L(x - 1, y; \sigma))^2 + (L(x, y + 1; \sigma) - L(x, y - 1; \sigma))^2}; 
   \]
   (25)
   \[
   \theta(x, y, \sigma) = \tan^{-1}\left(\frac{L(x, y + 1; \sigma) - L(x, y - 1; \sigma)}{L(x + 1, y; \sigma) - L(x - 1, y; \sigma)}\right); 
   \]
   (26)
2. Smooth magnitudes with a Gaussian window \( (1/2\pi\sigma)^{3/2} \exp(-(x-x_m)^2+(y-y_m)^2)/2\sigma^2 \), with the local scale parameter \( \sigma = 1.5\sigma_m \).
(3) each magnitude is accumulated to one of 36 predefined bins, $(\pi n)/10$ with $0 \leq n \leq 35$, if its corresponding orientation belongs to that bin. With this sub-step, a histogram with 36 bins is obtained;

(4) select the following orientations: (i) the orientation to the maximum of the histogram, and (ii) an orientation to a local maximum, whose value is above 80% of the maximum of the histogram;

(5) for each selected orientation, the following procedure is applied for orientation correction: (i) use a parabola to fit the histogram value of the orientation and those of its two neighbors; and (ii) replace the original orientation with the orientation corresponding to the peak position of the parabola; and

(6) create a keypoint $\hat{x}_k = [x_k, y_k, \omega_k]^T$ in the region, whose orientation is the same as one of the selected orientations. One or many keypoints will be created for a given extreme in this step and each keypoint has an orientation.

The regions detected by region detectors are invariant to more kinds of affine transformations than those detected by corner detectors, which ensures the high performance for image matching. Especially, the MSER, DoG operator, Harris–Laplace and Hessian–Laplace detectors are more suitable for detecting regions that are substantially brighter or darker than their surroundings. Among them, MSER is the optimal one in being able to deal with most varieties of transformations, including scaling, rotation, viewpoint variation, and illumination changes. Furthermore, when dealing with scaling, the SIFT detector, Harris–Laplace, and Hessian-Laplace are better choices than the others.

3. Feature description

Detection is followed by feature description. The simplest descriptor is a vector of pixel values. Ideal descriptors should be not only distinctive, i.e., they should be able to deal with a large number of objects and robust to occlusion and background clutter, but also only distinctive, i.e., they should be able to deal with a large number of the maximum of the histogram.

Planning. Existing descriptors can be categorized into the following five types: (i) filter-based descriptors [17,25,61], (ii) distribution-based descriptors [9,34,37,42,48], (iii) textons [40,81,82], (iv) derivative-based descriptors [23,36,63], and (v) others [12,79,80].

3.1. Filter-based descriptors

In this sub-section, we review filter-based descriptors, which are steerable filters [25], Gabor filters [17,18], and complex filters [61].

3.1.1. Steerable filters

The term “steerable filter” [25] describes a set of basis filters, which can synthesize any filter with an arbitrary orientation, i.e.,

$$F'(\theta) = \sum_{i=1}^{d} k_i(\theta) F_i,$$

where $F_i$ is the $i$th basis filter; $k_i(\theta)$ is the linear combination coefficient to synthesize the filter $F'$ with a given orientation $\theta$. A quadrature pair of filters, which means the two filters have identical frequency but one is the Hilbert transform of the other, can be applied to synthesize filters of a given frequency response with arbitrary phase. The derivatives of Gaussian have been demonstrated to be effective in many early vision and image processing tasks. The steerable quadrature filter pairs for the second order ($G_2$ and $H_2$), the third order ($G_3$ and $H_3$), and the fourth order ($G_4$ and $H_4$) of Gaussian are

\[
G_2(x, y) = 0.9213(2x^2 - 1) \exp(-(x^2 + y^2)),
\]

\[
H_2(x, y) = (-2.205x + 0.9780x^3) \exp(-(x^2 + y^2)),
\]

\[
G_3(x, y) = (2.472x - 1.648x^3) \exp(-(x^2 + y^2)),
\]

\[
H_3(x, y) = (-0.9454 + 2.959x^2 - 0.6582x^4)
\times \exp(-(x^2 + y^2)),
\]

\[
G_4(x, y) = (0.9344 - 3.738x - 1.246x^3)
\times \exp(-(x^2 + y^2)),
\]

\[
H_4(x, y) = (2.858x - 2.982x^3 + 3.957x^5) \exp(-(x^2 + y^2)),
\]

where $H_i$ is the Hilbert transform of $G_i$, for $i = 2, 3, 4$. Responses of these filters are shown in Fig. 6.

For $G_2$, it is not difficult to construct the basis filters according to Ref. [25] as

\[
G_{2a} = 0.9213(2x^2 - 1) \exp(-(x^2 + y^2)).
\]

Fig. 6. The upper three sub-figures from left to right are $G_2$, $G_4$, and $G_6$, respectively. The bottom three sub-figures from left to right are $H_2$, $H_4$, and $H_6$, respectively.
spatial bandpass functions, which have good spatial localization, Daugman [17,18] developed the 2D Gabor filters, a set of local receptive field profiles in the mammalian cortical simple cells. The visual cortex by Gabor filters, as they are similar to the applied for local feature representation within a pure computer standing and recognition. Consequently, Gabor filters have been decomposition are biologically relevant to human image under-sics research [44] suggest that Gabor filters based on image

3.1.2. Gabor filters

Research findings from cognitive psychology and psychophysics research [44] suggest that Gabor filters based on image decomposition are biologically relevant to human image understanding and recognition. Consequently, Gabor filters have been applied for local feature representation within a pure computer vision context.

Marcelja [44] and Daugman [17,18] modeled the responses of the visual cortex by Gabor filters, as they are similar to the receptive field profiles in the mammalian cortical simple cells. Daugman [17,18] developed the 2D Gabor filters, a set of local spatial bandpass functions, which have good spatial localization, orientation selectivity, and frequency selectivity. Lee [38] gave a good introduction to image representation using Gabor filters. A Gabor filter is the product of an elliptical Gaussian envelope and a complex plane wave, defined as

$$\psi_{x,y}(x, y) = \psi_{\delta}(\bar{z}) = \frac{||\bar{k}||}{\delta^2} \cdot \exp \left( - \frac{||\bar{k}||^2 \cdot ||\bar{z}||^2}{2\delta^2} \right) \times \left[ \exp \frac{\bar{k} \cdot \bar{z}}{\delta^2} - \exp \left( - \frac{\delta^2}{2} \right) \right].$$

(47)

where $\bar{z} = [x, y]$ is the variable in a spatial domain and $\bar{k}$ is the frequency vector, which determines the scale and orientation of Gabor filters, $\bar{k} = k_s e^{i\phi_s}$, where $k_s = k_{max}/f^s$, $k_{max} = \pi/2$, $f = 2$, $s = 0, 1, 2, 3, 4$, and $\phi_s = \pi d/8$, for $d = 0, 1, 2, 3, 4, 5, 6, 7$. Examples of the real part of Gabor filters are presented in Fig. 8, where Gabor functions (full complex functions) are with five different scales and eight different orientations, making a total of 40 Gabor functions. The number of oscillations under the Gaussian envelope is determined by $\delta = 2\pi$. The term $\exp(-\delta^2/2)$ is subtracted in order to make the kernel DC-free, and thus insensitive to the average illumination level.

3.1.3. Complex filters

Schaffalitzy and Zisserman [61] applied the following filters for local feature representation,

$$K_{mn} = (x + iy)^m(x - iy)^n G(x, y),$$

(48)
where $G(x,y)$ is a Gaussian; $m + n \leq 6$ and $m \geq n$. There are 16 different filter responses for a given image, as shown in Fig. 9. The responses of these filters: (i) are similar to derivatives of a Gaussian; (ii) are rotationally invariant; and (iii) act only on changing their relative phases but not the magnitudes.

3.2. Distribution-based descriptors

Distribution-based descriptors usually obtain better performance compared with other types of local descriptors. We review the following prevailing ones: the SIFT [42], the PCA-SIFT [34], the shape context [9], the spin image [37], the RIFT [37], and the GLOH [48].

3.2.1. Descriptor of sift

The SIFT descriptor [42] for each keypoint $\mathbf{z}_k$ (with scale $\sigma_k$ and orientation $\theta_k$) is a 128 dimensional vector created by first computing the gradient magnitude and orientation in the neighborhood of the keypoint. It contains 16 orientation sub-histograms, and each consists of 8 bins. In detail, we have the following steps (Fig. 10):

1. generate the center $\mathbf{c}_ij = [-6 + 4(i - 1), -6 + 4(j - 1)]^T$, where $1 \leq i,j \leq 4$, for each cell;
2. generate the location $\mathbf{l}_ij = [-8.5 + i, -8.5 + j]^T$, where $1 \leq i,j \leq 16$, for each point;
3. generate the orientation bin $\phi_i = (i - 5)\pi/4$, where $1 \leq i \leq 8$;
4. calculate the rotation matrix $R = \begin{bmatrix} \cos(\theta_k) & -\sin(\theta_k) \\ \sin(\theta_k) & \cos(\theta_k) \end{bmatrix}$;
5. rotate and translate centers and locations according to $\tilde{\mathbf{c}}_{ij} = R\mathbf{c}_ij + \mathbf{z}_k$ and $\tilde{\mathbf{l}}_{ij} = R\mathbf{l}_ij + \mathbf{z}_k$;
6. the gradient magnitude $m_{ij}$ and the orientation $\phi_{ij}$ of $\tilde{\mathbf{l}}_{ij}$ are sampled around the keypoint $\mathbf{z}_k$ on scale $\sigma_k$;
7. compute the x-coordinate weighting vector $\tilde{\omega}_{ij}^{(x)} \in \mathbb{R}^{16}$ of the location $\tilde{\mathbf{l}}_{ij}$ according to: $\tilde{\omega}_{ij}^{(x)} = \max(1 - |\tilde{e}_{mn}(1) - \tilde{e}_{mn}(1)|/4.0)|_{1 \leq m,n \leq 4}$, where $\tilde{e}_{mn}$ is the x-coordinate of $\tilde{e}_m$ and $\tilde{e}_n$. Compute $\tilde{\omega}_{ij}^{(y)} = \max(1 - |\tilde{e}_{mn}(2) - \tilde{e}_{mn}(2)|/4.0)|_{1 \leq m,n \leq 4}$. Construct the location weighting vector $\mathbf{w}_{ij}^{(h)} = [\tilde{\omega}_{ij}^{(x)}(m,n) \times \tilde{\omega}_{ij}^{(y)}(m,n)]_{1 \leq m,n \leq 4}$.

Repeat $\mathbf{w}_{ij}^{(h)} \in \mathbb{R}^{16} 8$ times to obtain $\tilde{\mathbf{w}}_{ij}^{(h)} \in \mathbb{R}^{128}$ because we have 8 orientations for each cell;
8. compute the orientation weighting vector $\mathbf{w}_{ij}^{(o)} \in \mathbb{R}^8$: (i) $\bar{\phi} = \text{mod}(\phi_{ij} - \theta_k - \bar{\phi} + \pi, 2\pi) - \pi$, where $\bar{\phi} = [\phi_i]_{1 \leq i \leq 8}$ and $\bar{\phi} \in \mathbb{R}^8$; (ii) $\mathbf{w}_{ij}^{(o)} = \max(1 - 4|\bar{\phi}/\pi, 0)$; and (iii) repeat $\tilde{\mathbf{w}}_{ij}^{(o)} \in \mathbb{R}^8 16$ times to obtain $\tilde{\mathbf{w}}_{ij}^{(o)} \in \mathbb{R}^{128}$ because we have 16 cells for each orientation;
9. the 128 dimensional histogram is $h_k = \sum_{ij=1}^{128} \tilde{\mathbf{w}}_{ij}^{(o)} \mathbf{w}_{ij}^{(h)} m_{ij}$, where $\mathbf{w}_{ij}^{(h)}$ is a Gaussian weighting factor. It is obtained by $\mathbf{w}_{ij}^{(h)} = \exp(-d_{ij}^2/(2 \times 8^2))/(2\pi \times 8^2)$, where $d_{ij}$ is the Euclidean distance between $\mathbf{z}_k$ and $\mathbf{l}_{ij}$; and normalization: (i) $h_k = h_k/\sum_{k=1}^{128} h_k(k)$; (ii) $\mathbf{h}_k = -\min(h_k, 0.2)$; and (iii) $\mathbf{h}_k = h_k/\sum_{k=1}^{128} h_k(k)$;
10. The calculated $\mathbf{h}_k \in \mathbb{R}^{128}$ is the descriptor of the keypoint $\mathbf{z}_k$ with scale $\sigma_k$ and orientation $\theta_k$.

3.2.2. PCA-SIFT

Ke and Sukthankar [34] proposed the PCA based SIFT (PCA-SIFT), which performs more efficiently than the SIFT descriptor for matching. The SIFT detector is applied here to detect keypoints. The PCA-SIFT descriptor is constructed with the following steps:

1. a $41 \times 41$ patch centered at each keypoint is extracted at a given scale and $N$ image patches are collected;
2. rotate each image patch according to its dominant orientation to a canonical direction;
3. for each image patch, the gradient maps computed both on orthogonal and vertical directions are combined together to generate a feature vector, which has $39 \times 39 \times 2$ elements;
(4) each feature vector is normalized to unit length to minimize the effects of illumination changes; and
(5) PCA is performed to reduce the feature dimension from 3042 to 20 based on the collected image patches.

Although the PCA-SIFT is simpler and faster than the SIFT, it loses some discriminative information during the process of dimension reduction.

3.2.3. Shape context
Shape context, a robust and simple algorithm to find correspondences between shapes, is a 3D histogram of edge point locations and orientations introduced by Belongie et al. [9]. Location is quantized into 9 bins using a log-polar coordinate system as displayed with the radius set to 6, 11 and 15 and orientation is quantized into 4 bins (horizontal, vertical and two diagonals), resulting in a 36 dimensional vector. It is similar to the SIFT descriptor, but is based on edges extracted by the Canny detector. The procedure to obtain shape contexts is as following:

(1) extract the shape contours of an image by an edge detector;
(2) sample $N$ points from each shape contour to represent the shape, each sampling point is considered as a reference point;
(3) for each reference point, construct a coordinate with the reference point as its origin;
(4) compare every reference point, e.g., the point $A$, with each of the other $N-1$ points, e.g., the point $B$, in the form of $\rho$ and $\theta$ as shown in Fig. 11, which results in a $N-1$ dimensional vector for each reference point;
(5) express each vector in the polar coordinate as the function of $\log \rho$ and $\theta$, where 5 bins are used for $\log \rho$ and 12 bins are used for $\theta$ to construct a two-dimensional histogram, as shown in Fig. 12; and
(6) count the number of points with the same $\log \rho$ and $\theta$ falling into the same bin.

In summary, the shape context describes the distribution of the rest points of the shape with respect to the reference point on the shape. Therefore, finding correspondences between shapes is equivalent to finding the point that has the most similar shape context on the other shape. In Ref. [9], shape contexts were applied for shape matching and object recognition.

3.2.4. Spin image
Spin image was first used in 3D object recognition by Johnson and Hebert [32], and later applied for planar images classification by Lazebnik et al. [37], where it is also called intensity-domain spin image, as it is a two-dimensional histogram representing the distribution of intensity in the neighborhood of a center point, whose two dimensions are $d$ and $i$, where $d$ is the distance from the center point and $i$ is the intensity value. The procedure is as following:

(1) extract a sparse set of affine-covariant elliptical regions from a texture image using the Harris affine or Laplacian blob detectors, which detect complementaty types of structures;
(2) normalize each elliptical region into a unit circle to reduce the affine ambiguity to a rotational one;
(3) before computing the spin images, slightly blur the normalized patches using a Gaussian kernel to reduce the potential sensitivity of the normalization to noise and resampling;
(4) quantize intensities of each normalized patch into 10 bins; and
(5) compute a 10 bin normalized histogram for each of 10 rings centered on the region, resulting in a $10 \times 10$ dimensional descriptor.

To achieve invariance to affine transformation of the image intensity function, i.e., $I \rightarrow al+b$, the intensity function range is normalized within the support region of the spin image (Fig. 13).

3.2.5. Rotation invariant feature transform
The RIFT was also proposed by Lazebnik et al. [37]. The major steps are as follows:

(1) extract a sparse set of affine-covariant elliptical regions from a texture image using the Harris affine or Laplacian blob detectors, which detect complementary types of structures;
(2) normalize each elliptical region into a unit circle to reduce the affine ambiguity to a rotational one;
(3) divide the circular normalized patch into concentric rings with equal width, where 4 rings are used; and
(4) compute a gradient orientation histogram with 8 orientations within each ring, resulting in a $4 \times 8$ dimensional descriptor.

To ensure rotation invariance, this orientation is measured at each point from a high intensity value pointed to a low intensity value, i.e. from a white region to a black region, as illustrated in Fig. 14.

However, the RIFT is sensitive to flip transformations of the normalized patch.
3.3. Textons

Textures can be characterized by their responses to a set of filters [16,40,62,81,82] and this Section reviews texton-based feature representations. Many object categorization algorithms [1,27,37,55] are relevant to texton models. There are usually four steps:

1. local appearance vectors generation: (i) collect n images, each with size \( M \times N \) for \( 1 \leq i \leq n \); (ii) select a set of filters, e.g., Gabor filters, as a filter bank; (iii) convolute each image with the filter bank and generate \( n \) filtered image sets; and (iv) cluster each image set based on a clustering algorithm, e.g., K-Means, into \( K_i \) cluster centers \( \hat{c}_{ij} \) in \( R^d \) for \( 1 \leq i \leq n \) and \( 1 \leq j \leq K_i \). These cluster centers are termed as appearance vectors. This procedure is shown in Fig. 15;

2. global appearance vectors clustering or dictionary generation: (i) collect all cluster centers \( \hat{c}_{ij} \); (ii) clustering these \( \sum_{i=1}^{n} K_i \) centers to \( K \) centers \( \hat{a}_i \in R^d \) with \( 1 \leq i \leq K \); and (iii) set \( \hat{a}_i \in R^d \) as initial cluster centers and apply a clustering algorithm to update these centers based on all vectors in the filtered image sets. The final \( K \) centers \( \hat{a}_i \in R^d \) with \( 1 \leq i \leq K \) are global appearance vectors or the dictionary. The procedure is illustrated in Fig. 16;

3. pseudo-image generation for representation: (i) generate a filtered image set for a given image based on the filter bank; and (ii) assign each vector a class label by comparing it with the global appearance vectors or dictionary according to the nearest neighbor rule. This procedure is shown in Fig. 17; and

4. texton generation: (i) stack each filter to a vector and combine all vectors as a matrix \( F \in R^{(d \times k) \times n} \); (ii) calculate the pseudo-inverse of \( F \) and multiply it with appearance vectors as texton vectors \( \bar{t}_i \in R^{(d \times k)} \) with \( 1 \leq i \leq K \); and (iii) unstack each texton vector \( \bar{t}_i \) to texton \( T_i \in R^{d \times k} \), a matrix. The procedure is shown in Fig. 18.

Different types of textons have been developed for feature characterization. The key issue for the texton construction is the filter bank selection. For example, Leung and Malik [40] selected 48 filters: first and second derivatives of Gaussians with 6 orientations and 3 scales, 8 Laplacian of Gaussian filters, and 4 Gaussians. They are shown in Fig. 19(a); Schmid [62] selected 13 rotationally invariant filters \( F(r, \sigma, \tau) = f_0(\sigma, \tau) + \cos(\pi r / \sigma^2) \exp(-r^2/2\sigma^2) \), as shown in Fig. 19(c); and Varma and Zisserman [81,82] developed the maximum response filter bank based on the root filter set, as shown in Fig. 19(b). This filter bank consists of 38 filters: a Gaussian filter, a Laplacian of Gaussian filter, an edge filter with 6 orientations and 3 scales, and a bar filter with 6 orientations and 3 scales. Different from other methods, Varma and Zisserman selected only one response from 6 orientations for both edge and bar filters in each scale, i.e., 8 responses are selected to characterize features.

3.4. Derivative-based descriptors

Local derivatives [36] were explored by Koenderink and Van Doorn. Florack et al. [23] incorporated a number of local derivatives and constructed the differential invariants, which are rotational invariant, for local feature representation.

One of the most popular derivative-based descriptors is the local grayvalue invariants developed by Schmid and Mohr [63], which are a set of differential invariants computed to third-order from the local derivatives. Let \( \bar{x} \) as an image and \( \sigma \) as a given scale.

The local derivative of order \( N \) at a point \( \bar{x} = (x_1, x_2) \) is \( L_{i_1 \ldots i_N}(\bar{x}, \sigma) \), which is obtained by convoluting \( I \) with Gaussian derivatives \( G_{i_1 \ldots i_N}(\bar{x}, \sigma) \), and \( I \) is \( |x_1, x_2| \).

Then the local grayvalue invariants can be described as \( F = \{ f_i \}_{1 \leq i \leq 9} \), where

\[
F_1 = L = L(\bar{x}, \sigma),
\]

(49)
\[ f_2 = L_i L_i = \sum_{i=1}^{2} l_i^2, \]
\[ f_3 = L_i L_j L_j = \sum_{i,j=1}^{2} L_i L_j L_j, \]
\[ f_4 = L_g = \sum_{i=1}^{2} L_i L_i, \]
\[ f_5 = L_i L_i = \sum_{i,j=1}^{2} L_i L_i L_i, \]
\[ f_6 = \frac{1}{C_1} \sum_{i,j,k=1}^{2} (L_i L_j L_i L_j L_k L_k - L_i L_j L_k L_i L_k L_k), \]
\[ f_7 = \frac{1}{C_0} \sum_{i,j,k=1}^{2} (L_i L_i L_j L_j L_k L_k - L_i L_j L_k L_i L_k L_k), \]
\[ f_8 = \frac{1}{C_0} \sum_{i,j,k,m=1}^{2} (L_i L_j L_k L_k L_m L_m - L_i L_j L_k L_k L_m L_m). \]
Fig. 17. Pseudo image generation for representation.

Fig. 18. Texton generation.

Fig. 19. Filter banks for texton-based feature representation: (a) Leung and Malik filter bank; (b) Varma and Zisserman filter bank; (c) Schmid filter bank.
\[ f_g = L_9l_4l_5 = \sum_{i,j,k=1}^2 \lambda_{31} \lambda_{31} \lambda_{31} \lambda_{31} \lambda_{31}. \]  
\[ (57) \]

where \( \lambda_{31} \) is the 2D anti-symmetric epsilon tensor defined by \( \varepsilon_{12} = -\varepsilon_{21} = 1 \) and \( \varepsilon_{11} = \varepsilon_{22} = 0 \).

The major steps to obtain the descriptor are: (i) interest points are extracted by the Harris corner detector; and (ii) a rotationally invariant descriptor is computed for each of the interest points, according to Eqs. (49)–(57).

3.5. Others

Apart from the four types of descriptors above, there are also other local features, e.g., the moment-based descriptor [80], the phase-based local features [12], and the color-based descriptors [79].

3.5.1. Moment-based descriptors

Generalized moment invariants [80] are computed up to second order and second degree for derivatives of an image patch: 
\[ M^p_{pq} = (1/\sqrt{\pi}) \sum_{x,y} \rho^p_{q} x^p y^q I(x,y), \]  
order \( p \) and degree \( q \). \( I_d \) is the image gradient in the direction \( d = x, y \).

3.5.2. Phase-based local features

Multi-scale phase-based local features [12] utilize both the local phase information and scale changes and are likely to provide improved invariance to illumination changes. Moreover, this type of feature descriptor outperforms others when applied to images that are subjected to illumination changes. Moreover, this type of feature description. They can deal with scale changes and are likely to provide improved invariance to illumination changes.

3.5.3. Color-based descriptors

Color-based local features are based on color information. Four color descriptors were proposed by Weijer and Schmid [79], which are histograms of rgb, hue, opponent angle and spherical angle. The first two descriptors called color invariants are based on zero-order invariants; while the others named as color angles are based on first-order invariants. To test their effectiveness against shape-based descriptors, e.g., the SIFT, they were applied for matching, retrieval, and classification. For color objects, a pure color-based approach achieves better performance than a shape-based one. In general, the combination of color and shape descriptors outperforms the pure shape-based approach. The color descriptors are given as following:

(1) for rgb, the normalized \( r \) can be described as
\[ r = \frac{R}{R + G + B}. \]  
\[ (63) \]

Normalised \( g \) and \( b \) have similar equations as Eq. (63). The normalized rgb can be considered to be invariant to lighting geometry and viewpoint;

(2) for hue
\[ \text{hue} = \arctan \left( \frac{O_1}{O_2} \right), \]  
\[ (64) \]

where \( O_1 = (R - G)/\sqrt{2} \) and \( O_2 = (R + G - 2B)/\sqrt{6} \) are opponent colors;

(3) for opponent angle
\[ \theta^0 = \arctan \left( \frac{O_1}{O_2} \right), \]  
\[ (65) \]
where \( (O_1)_y = (R_y - G_y) / \sqrt{2} \) and \( (O_2)_y = (R_y + G_y - 2B_y) / \sqrt{6} \) are the derivatives of the opponent colors. They are invariant to specular variations. The opponent angle is robust to geometric changes; and

(4) for spherical angle

\[
\theta^o = \arctan \left( \frac{(O_1)_y}{(O_2)_y} \right)
\]

(66)

where \( (O_1)_y = (G_y R - R_y G) / \sqrt{R^2 + G^2} \) and \( (O_2)_y = (R_y R B + G_y R B - B_y R^2 - B_y G^2) / \sqrt{(R^2 + G^2)(R^2 + G^2 + B^2)} \) are changes in the two directions perpendicular to the object color, which are invariant to geometric variations.

4. Conclusion

Because of the promising properties and capabilities of local features, they are being utilized broadly for various kinds of computer vision applications, e.g., object recognition, object class recognition, texture classification, image retrieval, robust matching and video data mining.

In this review, detectors are divided into corner detectors and region detectors. For example, corner detectors consist of Moravec’s corner detector, Harris corner detector, SUSAN, Trajkovic operator, and high-speed corner detector; and region detectors discussed are the Harris–Laplace, Harris affine and Hessian affine, edge-based region detector, intensity-based region detector, maximally stable extremal region (MSER) detector, salient region detector, difference of Gaussian (DoG) operator and scale invariant feature transform (SIFT) detector.

Local descriptors introduced were following five types: (i) filter-based descriptors, (ii) distribution-based descriptors, (iii) textrons, (iv) derivative-based descriptors, and (v) others. Further details were given on filter-based descriptors involving steerable filters, Gabor filters, and complex filters; distribution-based descriptors comprise SIFT descriptor, PCA based SIFT (PCA-SIFT), shape context, spin image, rotation invariant feature transform (RIFT), and gradient location and orientation histograms (GLOH); textrons contain Leung and Malik 3D textrons, Varma and Zisserman model, and Schmid filter bank; derivative-based descriptors consist of local derivatives, Florack descriptor, local grayscale invariants; others are general moment invariants, phase-based local features, and color-based local features.

According to current performance evaluation studies, we can make the following tentative conclusions for detectors and descriptors: (i) MSER combined with SIFT descriptor usually performs the best for flat images but poorly for 3D objects; (ii) Hessian affine and DoG combined with an arbitrary descriptor consistently outperform other approaches for 3D objects; and (iii) SIFT and GLOH are generally accepted as the most effective descriptors. However, new detectors and descriptors are constantly being reported, and some of these may surpass the performance of currently accepted standards and increasingly demonstrate the power of feature-rich portrayal of imagery.

References
